Evaluating Adaptive Cruise Control Strategies in Worst-Case Scenarios

van Willigen, W.H.; Schut, M.C.; Kester, L.J.H.M.

published in
14th IEEE Conference on Intelligent Transportation Systems (ITSC)
2011

Evaluating Adaptive Cruise Control Strategies in Worst-Case Scenarios

W.H. van Willigen
VU University Amsterdam (NL)
and TNO The Hague (NL)
w.h.van.willigen@vu.nl

M.C. Schut
VU University Amsterdam (NL)
m.c.schut@vu.nl

L.J.H.M. Kester
TNO
The Hague (NL)
leon.kester@tno.nl

Abstract— **This paper is concerned with safety in (cooperative) adaptive cruise control systems. In these systems, the speed of the cars is maintained automatically, based on the preferred speed of the driver and the speed of the preceding car. Technologies that are used in these systems, such as radar and radio communication, introduce many factors of uncertainty in the system. In this paper, we present models for different adaptive cruise control strategies, in which this uncertainty is explicitly modelled. By simulating emergency braking situations under these uncertain circumstances, we find the minimal safe time headway for these strategies.**

I. Introduction

Nowadays, many commercially available cars include adaptive cruise control (ACC) functionality. This extension to normal cruise control uses a radar to determine the distance and relative speed of the preceding vehicle, and controls the acceleration based on this information. This functionality increases comfort and safety of the driver.

However, traffic throughput is not necessarily improved: ACC-equipped vehicles still maintain a rather large following distance, which has two reasons: first, it is a convenience system, not a safety system, so the human driver should have enough time to correct errors made by the ACC, should they occur. Second, this distance is needed in order not to exhibit aggressive driving behaviour. Traffic throughput would be improved if cars would be able to drive closer to each other. To this end, the notion of cooperative adaptive cruise control (CACC) is introduced. This extension to ACC functionality includes direct radio communication between vehicles. This enables a car to directly communicate its change in acceleration to its predecessor, which leads to faster response times.

Current ACC-equipped vehicles maintain a time headway of at least 1 second to their preceding vehicle, which is about the same time headway as humans keep on highways. This distance is 'safe', in the sense that if a preceding car does an emergency brake, there is enough time for the radar and/or the driver in the following vehicle to react to this. However, in CACC-controlled vehicles, the time headway (in the order of $0.x$ seconds) is much too short for humans to react to emergency brakes. This means that drivers in CACC-equipped vehicles have to rely solely on the car's ability to detect emergency brakes and to react accordingly. Determining the minimal safe time headway is therefore essential for CACC-equipped vehicles.

In this paper, we experimentally determine the minimal safe time headway for three different controllers: the ACC controller, that uses radar technology to derive information about its preceding vehicle; the CACC1 controller, that communicates the value of the car's accelerometer to the following vehicle; the CACC2 controller, that has a built-in braking model, that estimates the change in acceleration directly after a braking action occurs, which is *before* the car starts decelerating. We test these controllers using different initial velocities and different initial distances between the cars.

In our models, different factors of uncertainty are introduced, such as uncertainty in accuracy of radar readings and uncertainty in communication success. The values for these uncertainties are based on realistic values that occur in currently used technology. This uncertainty makes this model realistic and therefore useful in practice.

Design of CACC is in general concerned with two main issues: *safety* and so-called *string stability*. Most current research is about the string stability issue, but the safety issue is largely ignored, especially when taken into account the uncertainty in information and communication that these systems have to deal with.

In this paper, we only focus on the safety aspect. Our safety controller should be able to react to an emergency brake by its preceding vehicle when necessary. The comfort controller is concerned with keeping a platoon of cars smooth (i.e. maintaining a steady velocity) and string stable. String stability in a platoon means that oscillations in speed within the platoon will be damped by the following vehicles instead of amplified.

The remainder of this paper is structured as follows. In Section II, we discuss related literature. In Section III, we introduce the model that we used for our research. Section IV describes the experiments and results. We analyse these results in Section V, and we conclude the paper in Section VI.

II. Related Work

In intelligent transportation systems (ITS), we can identify different branches of research: one is focused on developing intelligent vehicles in order to improve throughput in highway traffic, and the other is focused on including intelligence in road infrastructure to improve this. The different branches enable different applications. Smart infrastructure can enable traffic monitoring and is also able to improve traffic throughput by means of giving speed advice to the drivers, or directly to the adaptive cruise control.

Our interest lies in the development of intelligent vehicles. A first step into this direction is the development of adaptive cruise control, in which a car computes its distance to the preceding vehicle by means of a radar. Marsden *et al.* [5] provide a comprehensive article about this technology and its implications on traffic flow.

Taking this technology a step further, we come in de domain of cooperative adaptive cruise controls (CACC), in which direct communication between vehicles allow them to react faster to each other. Van Arem *et al.* [7] describe the effect of CACC on traffic flow. They conclude that, when the penetration level of CACC-equipped vehicles is high enough ($> 60\%$), traffic stability and throughput is improved. In Yang *et al.* [8], a communication protocol is proposed in order to make a cooperative collision warning system on highways.

The main application area of CACC technology these days is platooning. Broggi *et al.* [1] and Kanellakopoulos [3] both use image recognition techniques in combination with sensors to autonomously enable platooning. However, current technology has improved significantly since then, and nowadays direct radio communication between vehicles is used to enable platooning.

In [2], an extensive architecture is given for a layered multi-agent CACC architecture. The authors use this architecture to implement both centralised platoons (in which there is a coordinating platoon leader) and decentralised platoons (in which all cars operate as equals). Khan *et al.* [4] present different platoon (in their paper, convoy) forming strategies, based on a utility value of a platoon.

One of the missing elements of the above approaches to designing CACC systems, is that they do not explicitly account for uncertainty in information and communication. Machine learning techniques could be a promising new way of autonomously learning the uncertainty in the vehicles.

An important aspect of platooning is ensuring string stability within a platoon. A thorough control-theoretic model of string stability in CACC is presented by Naus *et al.* [6].

To conclude, current research is mainly about CACC and how to design and implement these systems. We found that the safety aspect of the problem, while taking uncertainty of information and communication into account, is not often considered, although it crucial to the eventual acceptation of such systems by the public. Therefore, this will be the main focus of this research.

III. Model

In this section, we describe the model that we used in our simulations. Since there are many sources for uncertainty in the model, a mathematical analysis of our model is complex. We also intend to use this model as a starting point for more experiments, that would include more complex behaviours and more uncertainty. This well justifies our choice for simulation.

A. Cars

We model three types of car: cars containing an adaptive cruise control (ACC) controller and two different types of cooperative adaptive cruise control (CACC1 and CACC2) controller. The main difference between the ACC controller and the two CACC controllers is that the ACC controller uses radar technology to derive information about the preceding vehicle, whereas the CACC controllers use direct communication to derive the same information. This difference in technology has some implications. First, radar can only derive information about distance and (relative) velocity. This means that information about the acceleration of a preceding car has to be derived from this information. Radio communication can be much more informative, because any information can be transmitted, if the car considers it useful. In our setting, cars only communicate their acceleration. Second, information that cars know about themselves is generally more accurate than information that is derived by a radar.

The controllers only control the acceleration of the vehicle. Since our experiments take place on a highway with only 1 straight lane, there is no need for incorporating steering or lane changing in our model.

There are properties that apply to both the simulation of the ACC and the two CACC controllers. They both share the same update scheme, that we use in our discrete time-based simulation. This scheme is depicted in Algorithm 1.

```
/* Δt = 0.01                                    */
foreach timestep t do
    ẍ_t ← compute new a;
    ẋ_t ← ẋ_{t−1} + ẍ_t Δt;
    x_t ← x_{t−1} + ẋ_t;
end
```

Algorithm 1: Updating scheme for cars

In this scheme, $\ddot{x}_t$ is the acceleration of the car at time t, $\dot{x}_t$ is the velocity of the car at time t, and x_t is the position of the car on the road at time t.

There is a small delay between a car's braking activity (i.e. pressing the brake pedal) and the actual deceleration of the car. This delay is 150ms. This means that when a car decelerates at $t = 0$ and the car behind hits the brakes at $t = 0.5$, the car will start decelerating at $t = 0.65$.

In this model, car are not allowed to brake harder than -9m/s^2, and they will always obey this law. The maximum deceleration for each car is -9m/s^2, but sometimes cars brake less hard than they think they do, due to mechanical limitations. See the section on uncertainty below for more details about this.

The difference between the ACC and the two CACC controllers lies in the updating rule for the acceleration. In the following sections, these updating rules are described in detail.

1) ACC controller: The ACC controller uses radar technology to detect the acceleration of the preceding vehicle. This radar receives measurements at 10Hz (i.e. 10 per second), and it takes an additional 5ms to process each measurement. The radar measurements consist of the distance to the preceding vehicle and the relative velocity to the preceding vehicle. The acceleration of the preceding vehicle can be computed according to two consecutive measurements of the relative velocity.

In Algorithm 2, the pseudocode for the simulation of the ACC controller is given. The delays in processing radar data and braking are hardcoded: for example, if a braking activity occurs, the change in acceleration is then explicitly scheduled for $t + 0.15$.

The radars operate asynchronously, which can be seen in the algorithm: a radar measurement is done when $(t +$ radarOffset$)\%0.1 \ == \ 0$, with radarOffset in the interval $[0, 0.09]$.

/* $\Delta t = 0.01$**s** ***/**
foreach *timestep t* **do**
  if $(t +$ *radarOffset*$)\%0.1 == 0$ **then**
    $\dot{x}_{relative} \leftarrow$ do radar measurement ;
    $\dot{x}_{preceding,now} \leftarrow \dot{x}_{relative} + \dot{x}_{self}$;
    schedule measurement processing for $t + 0.05$;
  end
  if *Scheduled measurement* $\dot{x}_{preceding,now}$ **then**
    $\ddot{x} \leftarrow \dot{x}_{preceding,now} - \dot{x}_{preceding,previous}$;
    Braking activity to achieve $\ddot{x}$;
  end
  if *Braking activity to achieve* $\ddot{x}$ **then**
    schedule change in $\ddot{x}$ for $t + 0.15$;
  end
  if *Scheduled change in* $\ddot{x}$ **then**
    change $\ddot{x}$;
  end
  update $\dot{x}$ according to $\ddot{x}$;
  update x according to $\dot{x}$;
end

Algorithm 2: The simulation of the ACC controller

2) CACC1 controller: Both CACC controllers use direct car-to-car communication to exchange messages containing the car's acceleration. These messages are sent by each car at a frequency of 10Hz. Sending a message has a delay of $1ms$.

The CACC1 controller sends the values from its accelerometer to the following vehicle. Since the delay from the radar processing is no longer present in this controller, this controller should be more responsive than the ACC controller.

In Algorithm 3, the pseudocode for the simulation of the CACC1 controller is given. The messages are sent asynchronously, which is hard-coded using a messageOffset in the interval $[0, 0.09]$.

/* $\Delta t = 0.01$**s** ***/**
foreach *timestep t* **do**
  if $(t +$ *messageOffset*$)\%0.1 == 0$ **then**
    messageBody = $\ddot{x}_{current}$;
    follower receives messageBody at $t + 0.01$;
  end
  if *Received message containing* $\ddot{x}$ **then**
    Braking activity to achieve $\ddot{x}$;
  end
  if *Braking activity to achieve* $\ddot{x}$ **then**
    Schedule change in $\ddot{x}$ for $t + 0.15$;
  end
  if *Scheduled change in* $\ddot{x}$ **then**
    change $\ddot{x}$;
  end
  update $\dot{x}$ according to $\ddot{x}$;
  update x according to $\dot{x}$;
end

Algorithm 3: The simulation of the CACC1 controller

3) CACC2 controller: The CACC2 controller is an extension to the CACC1 controller. This extension makes the CACC2 controller more responsive than CACC1. When a braking activity occurs in a car, it takes 150ms until the car actually decelerates. However, the CACC2 cars have a braking model inside, which estimates the actual deceleration of the car immediately after the braking activity. This means that a braking car can send a message containing an estimation of the deceleration *before* the deceleration actually happens. This results in much faster response times by the following vehicles.

In Algorithm 4, the pseudocode for the simulation of the CACC2 controller is given. The messages are sent asynchronously, which is hard-coded using a messageOffset in the interval $[0, 0.09]$.

The algorithm shows that the braking activity results in a change in acceleration after the mechanical delay of 150ms, but the message containing the modeled deceleration is scheduled right away, and sent to the following vehicle in the next message.

Note that in both CACC algorithms, we use time-based communication as opposed to event-based communication. We chose for time-based communication because it is more robust; a message is sent every 0.1s, instead of only when something of interest happens (e.g. an emergency brake). This means that cars can expect messages at a regular interval, so they can also notice if a message does not arrive.

4) Uncertainty in Information and Communication: The above algorithms are all not very complicated, and one could probably compute the minimum safety distance cars should keep from each other when driving. However, in reality, there is a lot of uncertainty in measurements and communication. The information that the cars use for their calculations are all based on uncertain values and measurements. For example, radars do not perfectly measure the relative speed to the

/* $\Delta t = 0.01$s */
foreach *timestep t* **do**
  if $(t + messageOffset)\%0.1 == 0$ **then**
    if *Scheduled message containing* $\ddot{x}$ **then**
      messageBody = $\ddot{x}$;
    else
      messageBody = $\ddot{x}_{current}$;
    end
    follower receives messageBody at $t + 0.01$;
  end
  if *Received message containing* $\ddot{x}$ **then**
    Braking activity with modeled $\ddot{x}$;
  end
  if *Braking activity with modeled* $\ddot{x}$ **then**
    Schedule change in $\ddot{x}$ for $t + 0.15$;
    Schedule message with $\ddot{x}$;
  end
  if *Scheduled change in* $\ddot{x}$ **then**
    change $\ddot{x}$;
  end
  update $\dot{x}$ according to $\ddot{x}$;
  update x according to $\dot{x}$;
end

Algorithm 4: The simulation of the CACC2 controller

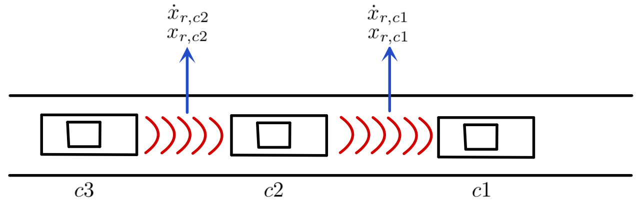

Fig. 1. The ACC scenario, in which cars $c2$ and $c3$ use their radars to derive the relative position $x_{r,i}$ and velocity $\dot{x}_{r,i}$ of vehicle i.

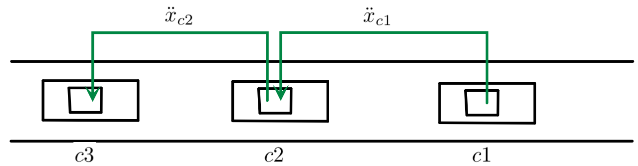

Fig. 2. The CACC scenario, in which vehicles i use radio to communicate their acceleration $\ddot{x}_i$ to their following vehicle.

preceding vehicle, it does contain some error. This type of uncertainty is *uncertainty in information*.

There is also *uncertainty in communication*: when a message is sent, there is no guarantee that the message will arrive.

We have modeled both these types of uncertainty. Below we describe each piece of uncertain information that we modeled. When we mention a value of σ, the uncertainty is normally distributed around the correct value μ: $\mathcal{N}(\mu, \sigma)$.

- Radar range rate: $\sigma = 0.1$m/s. This is the relative velocity measurement of the radar. This influences the computation of the preceding car's deceleration in the ACC vehicles;
- Failure in radar range rate: 1 in 1000. The radar fails to measure the relative velocity of the preceding vehicle in 0.1% of the cases;
- Own velocity: $\sigma = 0.1$m/s. This also influences the computation of the preceding car's deceleration in the ACC vehicles;
- Own max braking power: $\sigma = 0.3$m/s^2 one-sided. This is the error in a car's estimation of its own maximum braking power. For example, it could be that a car thinks it can brake with -9m/s^2, while in reality this is 8.7m/s^2. It is one-sided, since -9m/s^2 is a car's maximum braking power. This influences the CACC2 messages with the modeled braking power;
- Own modeled acceleration: $\sigma = 0.3$m/s^2. This is the uncertainty of the estimation of the acceleration when a braking action occurs. This is the value that is sent by the CACC2 vehicles, before the deceleration actually occurs;
- Own accelerometer value: $\sigma = 0.2$m/s^2. In the CACC1 controller, the car only sends out the estimation of its own acceleration. This has slightly less uncertainty than the modeled acceleration;
- Failure in broadcasting: 1 in 100. About 1% of sent messages do not arrive at its destination.
- Radar range measurement: $\sigma = 0.5$m. The distance to the preceding vehicle. This measurement is currently not used in our simulations, but becomes important when designing the controllers that have to keep a certain safe distance.

These values can be seen as realistic. However, different cars and technologies have different values of uncertainty. Of course, we can change these values for different cars. But then, the experiments would have to be done again as well.

IV. Experiments and Results

In this section, we will describe the experiments that we did and the results that we obtained.

A. Scenario

The objective of our experiments is to determine what the minimal safe time headway is between cars. Our experimental variable therefore is the time headway.

In our scenario, three cars drive on a highway, with varied initial velocities. The first car starts braking as hard as he can on $t = 0$. Then, we observe how the other cars react, and if any crashes occur. In Figures 1 and 2, we illustrate this scenario.

Using three cars in this scenario is sufficient for finding the minimal time headway . Using more cars in this scenario has no added value for the results. This is especially the case for the CACC controllers, when cars should be able to communicate their deceleration to more than one follower. While this is not a feature of CACC in this work, we do envision this functionality when further designing these controllers.

The first experimental variable is the type of controller: we did tests with an ACC controller and two different CACC controllers (CACC1 and CACC2); the second experimental variable is the time headway. We tested each value from 0.05s to 0.7s, with an interval of 0.01s. The third variable is the initial velocity of the cars, that we varied from 20m/s to 40m/s, with an interval of 5m/s. We ran each setting 50 times, resulting in a total of 3 controllers $\times$ 66 variations in time headway $\times$ 5 different initial velocities $\times$ 50 runs per setting $=$ 49.500 runs in total. The main observable is the cumulative number of crashes that occur in each setting.

B. Results

The results are summarised in Figures 3, 4 and 5. Figure 3 shows the results for ACC, Figure 4 shows the results for the CACC1 controller and Figure 5 shows the results for the CACC2 controller. From these graphs, it is apparent that the ACC controller performs the worst, the CACC1 controller is a bit better, and the CACC2 controller performs best.

V. Analysis

From Figures 3, 4 and 5, we can see that the CACC controllers outperform the ACC controller. This is what we expected. The difference can be attributed to the fact that the CACC controller can communicate its deceleration faster and more accurately. The ACC controller can only react on the actual deceleration of its predecessor, which makes it much slower. The CACC2 controller is faster than the CACC1 controller because it communicates its estimated deceleration, before the vehicle actually slows down.

It is nice to see that the number of crashes drops very steeply from a certain value. The crashes with the ACC controller drops between time headways between 0.4s and 0.5s, and the crashes with the CACC controller drops between time headways between 0.05s and 0.15s.

The most valuable values of these charts are, for each velocity, the lowest time headway at which no crashes occur. These values are plotted in Figure 6. This graph can now be used inside the controllers, to determine the preferred time headway to a preceding vehicle. Because the uncertainty in our models did introduce some outliers (see, for example, a crash that occurred with the CACC1 controller at $t = 0.61$ in Figure 4), we left out the most distant 2% of these outliers in this figure. The implication of leaving out these outliers is that, when using Figure 6 as a guideline for minimal safety distance, in approximately 2% of all emergency braking situations, a (very soft) crash occurs. We argue that this figure is acceptable, especially when keeping in mind that this figure could have been lower if we would have done more runs.

It may be interesting to compare these numbers with the official guideline. In the Netherlands, the guideline is to keep a time headway of two seconds to the preceding vehicle. However, a comparison with this guideline is a bit unfair, because nobody actually obliges this guideline. As

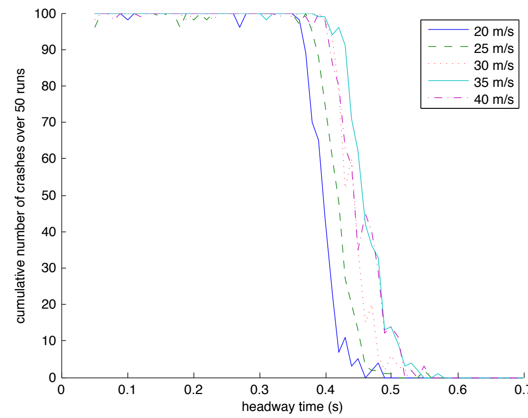

Fig. 3. Number of crashes vs. time headway for the ACC controller.

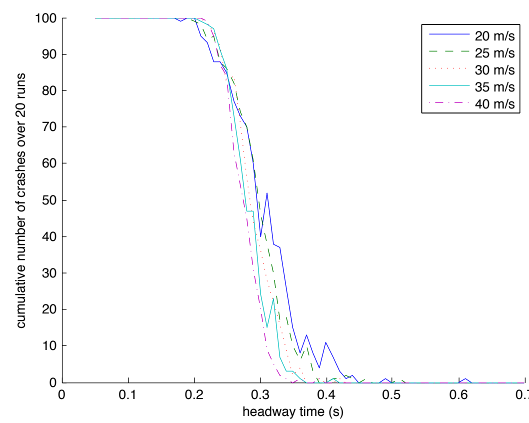

Fig. 4. Number of crashes vs. time headway for the CACC1 controller.

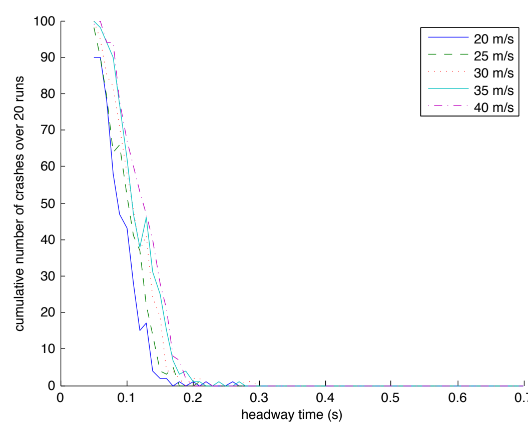

Fig. 5. Number of crashes vs. time headway for the CACC2 controller.

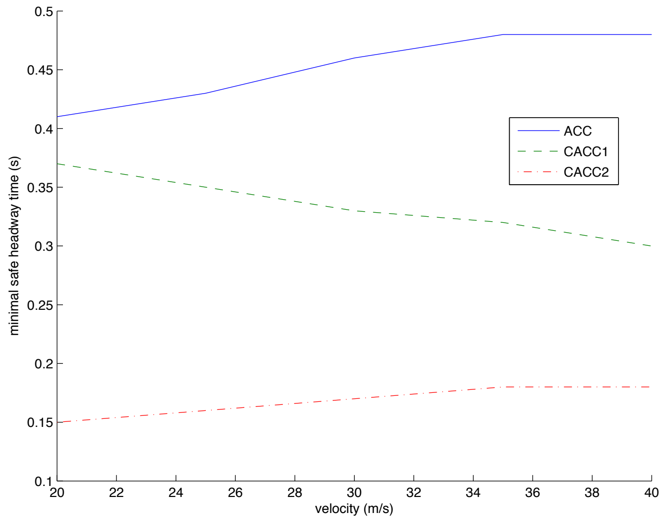

Fig. 6. Safe time headway vs. velocity.

stated in the introduction, current commercially available ACC systems have a minimal time headway of 1 second. When we compare this number to the results from figure 6, we see that using a CACC controller, the safe time headway is drastically improved.

The time headways that we found are based on our simulation results, which performed under very specific values of parameters in our model and the uncertainties. This means that when the car model or the uncertainty changes, we would have to rerun the experiments.

We will try to overcome this problem in future work. Ideally, our system would run on-line in a vehicle and is able to adapt to changing situations, such as different weather conditions and malfunctioning sensors. And instead of determining the number of crashes given some time headway, we will approximate the minimal safe time headway given the uncertain parameters and some desired maximum probability that a crash occurs. This will give our work a better theoretical and statistical foundation.

VI. Conclusions and Future Work

Conclusions

This paper describes work into the safety aspect of adaptive cruise control (ACC) and cooperative adaptive cruise control (CACC) strategies, taking into account the uncertainty under which these strategies must operate. We have modeled different strategies, including uncertainty of information and communication in the model. Because we included many different kinds of uncertainty that cars are encountering in real life as well, we can use the found values for safe time headway in the actual implementations of the controllers.

Our experimental results clearly show that the CACC controllers perform better than the ACC controller. We have also shown that the CACC2 controller, that is able to communicates an estimate of its deceleration *before* the vehicle actually decelerates, performs much better than the CACC1 controller, that is only able to communicate its measured deceleration.

Future Work

Currently, we have only tested our controllers in worst-case scenarios. This means that our controller can be considered only as a safety controller, that knows what the minimal time headway should be given a certain velocity and given uncertainty in parameters.

In the work of Naus *et al.* [6], a string stability controller is developed, in which the safety aspect is currently ignored. Therefore, we plan to combine our safety controller with their string stability controller.

Another thing we are planning to do is to do more experiments with the uncertainty in the controllers. Currently, values of the uncertainty are explicitly known to the controllers. However, in many cases, this uncertainty is unknown beforehand or can change over time. We will look at ways in which a controller is able to *learn* the correct values on the fly. We need on-line machine learning techniques for this, because these values are different for each vehicle, and they may even vary on-line as well. This means that a controller must learn these values for each new car that it encounters.

References

[1] A. Broggi, M. Bertozzi, A. Fascioli, C. G. L. Bianco, and A. Piazzi. Visual perception of obstacles and vehicles for platooning. *IEEE TRANS. INTELL. TRANSPORT. SYS*, 1(3):164–176, 2000.

[2] S. Hallé. Automated highway systems: Platoons of vehicles viewed as a multiagent system. Master's thesis, Université Laval, Québec, 2005.

[3] I. Kanellakopoulos, P. Nelson, and O. Stafsudd. Intelligent sensors and control for commercial vehicle automation. *Annual Reviews in Control*, 23:117 – 124, 1999.

[4] M. Khan, D. Turgut, and L. Bölöni. A study of collaborative influence mechanisms for highway convoy driving. In *Proceedings of International Workshop on Agents in Traffic and Transportation (ATT08), in conjunction with the Seventh Joint Conference on Autonomous and Multi-Agent Systems (AAMAS 2008)*, pages 46–53, May 2008.

[5] G. Marsden, M. McDonald, and M. Brackstone. Towards an understanding of adaptive cruise control. *Transportation Research Part C: Emerging Technologies*, 9(1):33 – 51, 2001.

[6] G. Naus, R. Vugts, J. Ploeg, M. van de Molengraft, and M. Steinbuch. String-stable cacc design and experimental validation: A frequency-domain approach. *Vehicular Technology, IEEE Transactions on*, 59(9):4268 –4279, 2010.

[7] B. van Arem, C. van Driel, and R. Visser. The impact of cooperative adaptive cruise control on traffic-flow characteristics. *Intelligent Transportation Systems, IEEE Transactions on*, 7(4):429 –436, 2006.

[8] X. Yang, J. Liu, F. Zhao, and N. H. Vaidya. A vehicle-to-vehicle communication protocol for cooperative collision warning. *Mobile and Ubiquitous Systems, Annual International Conference on*, 0:114–123, 2004.